A Causal Bayesian Network view of Reinforcement Learning

Charles Fox
Robotics Research Group
Engineering Science
University of Oxford

Neil Girdhar
Google

Kevin Gurney
Adaptive Behaviour Research Group
Department of Psychology
University of Sheffield

Abstract

Reinforcement Learning (RL) is a heuristic method for learning locally optimal policies in Markov Decision Processes (MDP). Its classical formulation (Sutton & Barto 1998) maintains point estimates of the expected values of states or state-action pairs. Bayesian RL (Dearden, Friedman, & Russell 1998) extends this to beliefs over values. However the concept of values sits uneasily with the original notion of Bayesian Networks (BNs), which were defined (Pearl 1988) as having explicitly causal semantics. In this paper we show how Bayesian RL can be cast in an explicitly Bayesian Network formalism, making use of backwards-in-time causality. We show how the heuristic used by RL can be seen as an instance of a more general BN inference heuristic, which cuts causal links in the network and replaces them with non-causal approximate hashing links for speed. This view brings RL into line with standard Bayesian AI concepts, and suggests similar hashing heuristics for other general inference tasks.

Introduction

Reinforcement Learning

An MDP is a tuple (S, A, p_s, p_r) where $s \in S$ are states, $a \in A$ are actions, $p_s(s'|s,a)$ are transition probabilities and $p_r(r|s,a)$ are reward probabilities. The goal is to select a sequence of actions $\{a_t\}$ (a *plan*) over time t to maximise the expected value $\langle v_t \rangle = \langle \sum_{t=1}^{T} \gamma^t r^t \rangle$, where T may be infinite, and each action is selected as a function of the current, observable state $a_t = a_t(s_t)$. We consider the case where p_s and p_r are unknown. Classical Reinforcement Learning approximates the solution using some parametric, point estimate function $\hat{v}(s, a; \theta)$ and seeks $\hat{\theta}$ to best approximate

$$\hat{v}(s_t, a_t; \hat{\theta}) \approx \max_{a_{t+1:T}} \langle \sum_{\tau=t}^{T} \gamma^\tau r_\tau \rangle = \langle r_t + \max_{a_{t+1}} \hat{v}(s', a; \theta) \rangle.$$

It runs by choosing a_t at each step (which may be $a = \arg\max_a \hat{v}(s, a)$ if best available performance is required; or randomised for ad-hoc exploratory learning), then observing the resulting r_t and s_{t+1} and updating θ towards a minimised error value (with $w_0 + w_1 = 1$):

$$\theta \leftarrow w_0\theta + w_1 \arg\min_{\theta'} (\hat{v}(s_t, a_t; \theta')) - [r_t + \gamma \max_{a_{t+1}} \hat{v}(s, a; \theta)])^2$$

Bayesian RL uses a larger parameter set ϕ to parametrise and learn a full belief over values $Q(v|s, a; \hat{\phi}) \approx P(v|s, a)$. (So classical RL is a special case where this parametric probability function is assumed to be a Dirac Delta function, $Q(v|s, a; \phi) = \delta(v; \hat{v}(s, a; \theta)))$

Causal Bayesian Networks

A Directed Graphical Model (DGM) is a set of variables $\{X_i\}$ with directed links specified by parent functions $\{pa(X_i)\}$, and a set of conditional probabilities $\{P_i(X_i|pa(X_i))\}$ so the joint is $P(\{X_i\}) = \prod_i P(X_i|pa(X_i))$. A Causal Bayesian Network (CBN) is a DGM together with a set of operators $do(X_i = x_i)$ which when applied to the model, set $pa(X_i) = \emptyset$ and $P(X_i) = \delta(X_i; x_i)$. The do operators correspond (Pearl 2000) to the effects of performing an intervention on the system being modelled. A DGM of a system 'respects causal semantics' if its corresponding CBN faithfully models interventions. (The name 'Bayesian Networks' originally referred (Pearl 1988) to CBNs.)

While DGMs and CBNs are generally treated as complementary, we will show how a hybrid net with some causal and some acausal links is a useful way to think about Reinforcement Learning algorithms, and suggests generalisations for creating other approximate inference algorithms.

The model

Perceived backwards-in-time causality is possible

The general MDP problem can be drawn as the CBN shown in fig. 1(a). Each (s, a) pair causes a reward and the next state. The task is to choose a plan to maximise the expected value. Classical AI tree-search methods ignore the v nodes and literally search the whole space of plans, computing the expected summed discounted rewards for each plan. This is generally intractable. (Exact polynomial dynamic programming can be used for cases where the set of possible states is the same at each step, but the polynomial order is the number of dimensions of s which may be large, rendering exact solution impractical though technically tractable.)

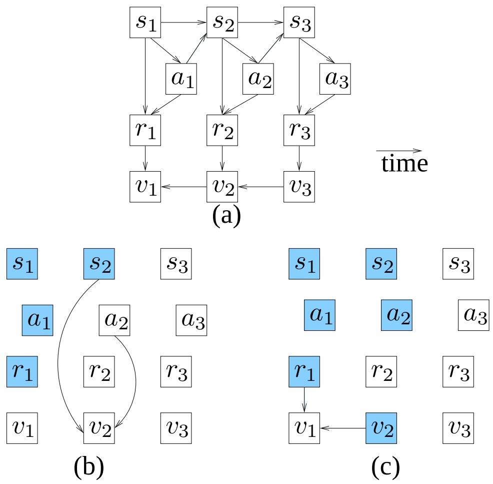

Figure 1: (a) Full inference. (b) RL action. (c) RL learning.

In the unknown p_s, p_r case they must also infer these probabilities from the results of their actions. Tree-searching can equivalently be conceived of as using the deterministic $v_t = \gamma v_{t+1} + r_t$ nodes to perform the summation. Inference proceeds as before: at each time t, for each plan (ordered by the classical search algorithm), we instantiate the plan nodes and infer the distributions over the v nodes, then perform the first action from the plan with the highest $\langle v_t \rangle$. Using message-passing algorithms (Pearl 1988), information must propagate all the way from the left to the right (via the a nodes) of the network and back again (via the v and r nodes) to make the inference. This is time-consuming.

The network respects causal semantics, and despite this it includes arrows pointing backwards through time. This is not a paradox: the expected discounted reward v_t at time t really is caused by future rewards: that is its meaning and definition. Note that v_t does not measure anything in the physical world, rather it is a 'mental construct'. So there is no backwards causation in the *physical* world being modelled. But there *is* backwards causation in the perceptual world of the decision maker: its percepts of future rewards cause its percept of the present v_t. For example, if v_t is today's value of a financial derivative, and we are able to give a delayed-execution instruction now that will somehow guarantee its value next month, then the causal do semantics would give a faithful model of this future intervention and an accurate current value. However value is a 'construct' rather than an object in the physical world, so no physical causality is violated.

Approximation method

We now present RL as a heuristic method for fast inference and decision making in the previous CBN. The two steps of RL, action selection and learning, (also known as the 'actor' and 'critic' steps) correspond to two DGMs derived from the CBN. We change the computation of the v nodes, from being exact to approximate expected discounted rewards. (Their semantics are the same, they still compute value, but their computational accuracy changes.)

In the action selection network shown in fig. 1(b), we sever all the causal links and replace them with two acausal links. So we assume an approximate, parametric distribution $Q(v_t|s_t, a_t; \phi)$. This is not a causal link: rather it is some function, whose parameters are to be learned, that will approximate the true causal $P(v_t|s_t, a_t, \hat{a}_{t+1}, ...\hat{a}_{t+N})$ where $\hat{a}_i$ are the actions in the optimal plan given a_t. It can be thought of as a 'hashing' link, making the best available guess under limited computational resources.

Once a_t is thus determined (though not yet executed) the learning step is performed. We remove the hashing links and reinstate the two causal links required to exactly update the parameters of v_t, shown in fig. 1(c). We assume that the best action was selected and will result in v_t from the action-selection step (which may be a Delta spike in classical RL or a belief distribution in Bayesian RL). The previous step's reward r_{t-1} is already known, so v_{t-1} is updated accordingly, using the standard Classical and Bayesian learning methods described earlier.

Following the learning step, the results of a_t are observed (r_t and s_{t+1}) and the actor phase begins again for $t+1$.

Discussion

Exact inference for action selection in the exact CBN is time-consuming, requiring inference across the complete temporal network and back again to evaluate each candidate in a plan searching algorithm. We have seen how RL can be viewed as a heuristic which cuts the causal links of the CBN and replaces them with acausal hashing links, mapping directly from the current state to an approximate solution. We suggest that having viewed RL in the context of CBNs, a similar method could be applied to more general inference problems when cast as CBNs: isolate the time-consuming parts of the network and replace them with trainable acausal hashing functions. The process of switching between causal and hashing links can also be seen in the Helmholtz machine (Dayan *et al.* 1995) in a similar spirit to RL.

We also saw how the causal do semantics allow for coherent backwards-in-time causality, where the caused entities are perceptual 'constructs' rather than physical entities. This makes an interesting contrast to other conceptions of causality such as (Granger 1969) which assume that causality must act forwards in time.

References

Dayan, P.; Hinton, G. E.; Neal, R.; and Zemel, R. S. 1995. The Helmholtz machine. *Neural Computation*.

Dearden, R.; Friedman, N.; and Russell, S. J. 1998. Bayesian Q-learning. In *AAAI/IAAI*, 761–768.

Granger, C. W. J. 1969. Investigating causal relations. *Econometrica* 37:424–438.

Pearl, J. 1988. *Intelligient Reasoning with Probabalistic Networks*. Morgan Kaufmann.

Pearl, J. 2000. *Causality*. Cambridge University Press.

Sutton, R., and Barto, A. G. 1998. *Reinforcement Learning*. MIT.